高技术通讯 2023 年 第33 卷 第10 期:1020-1028

doi:10.3772/j.issn.1002-0470.2023.10.002

基于注意力机制的对抗性协同过滤推荐算法①

吴哲夫② 程界斌 方路平③

(浙江工业大学信息工程学院 杭州 310023)

摘 要 针对协同过滤推荐算法中用户所交互的物品对其决策的不同贡献度问题,提出了一种基于相关注意力的协同过滤推荐算法。该算法结合深度学习中的注意力机制为不同物品分配不同的权值来捕获与目标物品最相关的物品,探索不同物品的权重对模型预测的影响并以此提升推荐的准确度;在此基础上,为了解决推荐算法鲁棒性低的问题,进一步提出了注意力协同对抗性训练的推荐算法,通过对抗性学习的方法并使用快速梯度符号算法(FGSM)构建对抗样本输入模型进行对抗训练,缓解模型受扰动的影响从而提升算法鲁棒性。在 Pinterest 和 MovieLens-1M 这 2 个数据集上的实验结果表明,所提算法不仅有效提升了推荐算法的准确度,同时也增强了推荐系统的鲁棒性。

关键词 协同过滤; 注意力机制; 对抗性学习; 鲁棒性

0 引 言

在信息爆炸的时代,推荐系统在诸如电子商务、社交网络、新闻门户等面向用户的在线服务中发挥着愈来愈重要的作用。一个优秀的推荐系统不仅可以帮助用户快速获取所需信息,还可以建立客户忠诚度,增强其对公司的粘性。推荐系统已成为信息检索和数据挖掘的重要手段之一[1]。

基于物品的协同过滤(item based collaborative filtering, ICF)是当前推荐算法中最主要的一种[2],其原理是通过“用户-物品”的交互数据来预测用户的偏好并生成 K 个用来推荐给用户的候选物品,然后再通过一个排序系统对生成的候选进行 Top-K 排序[3]。早期的 ICF 使用统计度量来量化 2 个物品间的相似性[4],例如物品间的余弦相似性,但为了使模型表现良好,此类方法通常需要对相似性度量进行手动调整。近年来,研究人员进一步地通过数据驱动的方法来学习物品间的相似性,代表性的 2 种方法是稀疏线性模型(sparse linear methods, SLIM)[5]和物品相似性分解模型(factored item similarity methods, FISM)[6]。SLIM 通过对数据的稀疏性和非负性限制,直接学习物品-物品的相似度矩阵,而 FISM 则将 2 个物品间的相似度分解为它们的潜在向量的内积。上述方法只考虑了物品间的简单交互而没有考虑到高阶交互关系,例如某些具有共同属性的多个物品。为了有效捕捉物品间的高阶交互关系,Feng 等人[7]提出了一种深度协同过滤推荐算法(deep item-based collaborative filtering for top-n recommendation, DICF),该算法使用不同的神经层来分别捕获物品间的低阶和高阶关系。具体地说,首先在神经网络的底层,通过对每个物品对的嵌入进行 Hadamard 乘积运算来捕捉物品间的低阶交互关系;然后,使用一个深度交互层,并利用非线性学习的方法来捕捉物品间的高阶交互关系[8]。得益于多层神经网络强大的学习能力,这种端到端的方案可以有效捕捉到物品间高阶关系对用户决策的影响。

① 浙江省自然科学基金重点项目(LZ22F010005)资助。
② 男,1971 年生,博士,副教授;研究方向:通信网络,数据挖掘;E-mail: wzf@zjut.edu.cn。
③ 通信作者,E-mail: flp@zjut.edu.cn。
(收稿日期:2022-03-14)

尽管上述方法在推荐准确性上取得了良好的性能,但还是忽略了历史上不同交互物品对用户当前选择贡献度的差异。事实上,某个用户对其所交互过的一组历史物品中不同物品的偏好应该是不同的。例如,用户分别购买了衣服、电脑和皮包3种物品,但并不意味着用户对这3种物品具有相同的偏好。当用户需要购买一件上衣时,显然衣服这个历史样本对当前决策的影响更大,因此它的权重因子也就更大。为了处理这类问题,引入深度学习中的注意力机制[9],其核心思想是从原始信息中捕获与当前任务最相关的部分并进行加权处理。通过注意力机制,为不同的历史物品分配不同的权重,从而提升模型的推荐精度。

此前的研究表明[10],如果在训练过程中人为地对模型输入施加微小的扰动时,模型性能会急剧下降,这也就意味着推荐模型抗干扰能力差。受到计算机视觉领域对抗性机器学习的启发[11],本文发现将对抗性学习结合到推荐系统模型中时,可以有效增强模型的鲁棒性。通过对用户和物品嵌入对抗扰动,再将扰动后的样本和原始样本一起输入到模型中进行训练,可以有效提升模型的鲁棒性。

本文的主要工作如下:(1)将注意力机制与协同过滤推荐算法相结合提出了基于注意力机制的协同过滤推荐算法(relevant attention-based collaborative filtering, RACF),解决用户交互物品的不同贡献度问题;(2)在此基础上,提出了一种基于注意力机制的对抗性协同过滤推荐算法(adversarial collaborative filtering recommendation algorithm based on relevant attention, ARACF),通过对抗性学习的方法来增强推荐算法的鲁棒性;(3)在MovieLens-1M和Pinterest这2个数据集上进行实验,验证了所提算法的有效性。

1 相关工作

1.1 注意力机制

神经网络中的注意力机制是在众多的输入信息中聚焦于对当前任务更为关键的信息,降低对其他信息的关注度,甚至过滤掉无关信息,进而提高任务处理的效率和准确性。近年来,基于注意力机制的推荐模型快速发展并取得了突出表现[12]。Xiao等人[13]使用注意力对模型中的每个交叉特征进行加权,缓解无用特征的影响,增强了模型预测精度。Zhou等人[14]使用注意力机制,提出了抑制过拟合的自适应正则化,针对特征出现的频率,来自适应调整它们的正则化强度。Seo等人[15]在卷积神经网络中使用双重注意力对用户偏好进行建模。Yi等人[16]使用注意力对用户-物品的评分重要性进行加权,值得注意的是,他们通过相似度矩阵来计算权重。He等人[17]提出了一个基于神经注意力物品相似性模型,引入注意力机制给不同物品加权,对物品-物品之间的相似性进行建模。Chen等人[18]提出了一个注意力驱动的推荐模型,使用注意力机制来估算用户对不同物品项目特征的分布,提高模型的可解释性。上述方法都聚焦于提升模型的准确度且取得了不错的效果,但却没考虑到模型的鲁棒性。基于深度学习的模型本身就十分脆弱,抵御外界攻击的能力较弱,因此对其鲁棒性的研究必不可少。本文借鉴对抗学习的思路,对模型进行对抗训练,从而增强推荐系统的鲁棒性。

1.2 对抗性训练

对抗训练是提升机器学习模型鲁棒性的重要方式,其原理是在模型训练过程中动态地生成对抗样本,将其添加到模型中进行训练,使得模型能够适应这些样本,进而提升鲁棒性。推荐系统的本质也是一个分类任务[19],因此对抗性训练的方法同样适用于推荐系统。He等人[20]在对抗性个性化排序系统中首次利用对抗训练,他们在用户-物品的向量嵌入上添加对抗扰动,验证了对模型鲁棒性的有效作用。Tang等人[21]考虑了视觉特征对推荐模型的影响,利用卷积神经网络提取图片特征并将对抗扰动添加在图片的潜在特征上。Park等人[22]提出了一种基于隐式反馈的推荐系统,该系统分别对正负样本进行对抗训练,实验还证明了对抗训练方法对离散输入和连续输入都有效。Yuan等人[23]提出了对抗训练协同去噪自编码器的推荐系统,将对抗扰动添加在编码器和解码器的参数上,并使用一个对抗训练框架进行训练。此外,他们还考虑了对抗扰动更细

粒度的影响[24],除了编解码器,他们还在用户的嵌入矩阵和模型的隐藏层上添加了对抗扰动。上述方法都有效增强了推荐系统的鲁棒性。对抗训练的关键在于如何构建对抗扰动,现有的扰动构建方式都或多或少存在一些问题,比如,快速梯度符号算法(fast gradient sign method, FGSM)存在线性假设问题(损失函数是线性的或者至少是局部线性的,否则梯度提升的方向就不一定是最优方向)、投影梯度下降(projected gradient descent, PGD)算法浪费计算资源等。因此,还需探索更优良的扰动构建算法。

2 模型架构

本节详细介绍基于注意力网络的协同过滤推荐模型(RACF),并在此基础上利用对抗性学习,提出了注意力协同对抗性训练的推荐算法(ARACF),以提升模型的鲁棒性。图 1 是所提模型框架图。

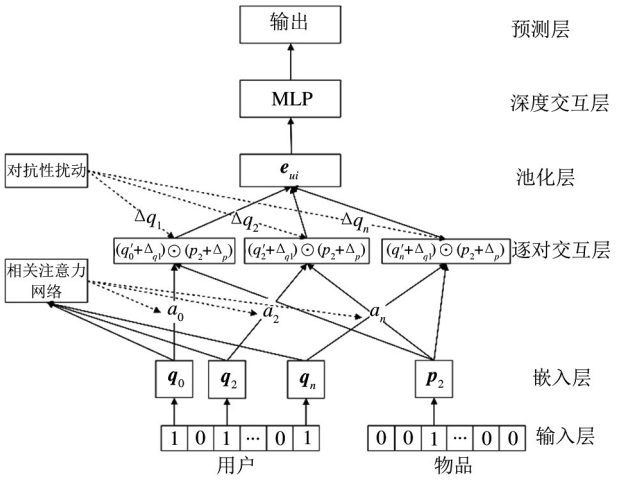


图 1 模型框架图

2.1 RACF 模型

为了区分用户不同交互物品的贡献度,本文采用深度学习中的注意力机制,提出了基于注意力网络的协同过滤推荐算法(RACF),整个模型包括输入层、嵌入层、注意力层、逐对交互层、池化层、深度交互层以及预测层。

在输入层,分别使用 one-hot 和 multi-hot 编码来对目标物品 i 和用户的历史物品集 R_u^+ 进行特征嵌入,将目标物品和每一个历史物品 $j \in R_u^+$,映射成一个维度为 16 的特征向量 $\boldsymbol{p}_i$、$\boldsymbol{q}_j$。因此,嵌入层输出的就是代表用户物品的特征向量 $\boldsymbol{q}_j$ 的集合和代表目标物品的特征向量 $\boldsymbol{p}_i$。

在嵌入层之上,为了获取用户对不同历史交互物品的偏好,通过一个注意力网络对用户嵌入的特征向量分配不同的权重,计算方法如下:

$$z_j = \boldsymbol{h}^{\mathrm{T}}\mathrm{ReLU}(\boldsymbol{W}(\boldsymbol{q}_j) + \boldsymbol{b}) \tag{1}$$

$$a_j = \exp(z_j)/\sum_{j\in R_u^+}\exp(z_j) \tag{2}$$

$$\boldsymbol{q}_j' = a_j \cdot \boldsymbol{q}_j \tag{3}$$

其中,$\boldsymbol{W}$ 和 $\boldsymbol{b}$ 分别是将输入映射到隐层的权值矩阵和偏置向量;$\boldsymbol{h}^{\mathrm{T}}$ 是将隐层映射到输出注意力权值的映射向量;a_j 是得到的用户对其交互过的历史物品 j 的偏好。

为了捕捉目标物品和用户历史物品间的二阶关系,对它们的嵌入进行 Hadamard 乘积运算,即 $\boldsymbol{q}_j'\odot\boldsymbol{p}_i$。此外,为了研究模型的鲁棒性,对 $\boldsymbol{q}_j'$ 和 $\boldsymbol{p}_i$ 分别施加对抗扰动 Δ_q、Δ_p,通过调整对抗性学习的参数 ε 直至找出最佳的扰动大小,对抗性扰动的构建由 2.2 节给出。此时交互层的 Hadamard 乘积运算变为

$$V_{ui} = (\boldsymbol{q}_j' + \Delta_q)\odot(\boldsymbol{p}_i + \Delta_p) \tag{4}$$

由于不同用户的历史物品的数目可能不同,因此逐对交互层的输出将没有固定的大小。池化层的作用就是将不同的输出转变为一个固定大小(k = 16)的向量,便于下一步的处理。在这里,使用加权平均池化,其输出一个向量:

$$\boldsymbol{e}_{ui} = \frac{1}{(|R_u^+|-1)^a}\left(\sum_{j\in R_u^+\setminus i}(\boldsymbol{q}_j' + \Delta_q)\odot(\boldsymbol{p}_i + \Delta_p)\right) \tag{5}$$

其中,a 是归一化超参数,用来平滑不同大小的 V_{ui}。当 a = 1 时代表没平滑,即为标准池化;a = 0 时,即为标准和池。由于数据集的不同,a 的大小并没有一个固定值。

为了探索物品间的高阶交互关系,通过一个 3 层感知网络(MLP)来对池化层的输出 $\boldsymbol{e}_{ui}$ 进行建模,将 $\boldsymbol{e}_{ui}$ 通过逐层传递,每一层赋予不同权值和偏置值,用以捕捉物品嵌入特征间的高阶关系。其隐藏层网络结构如下:

$$\begin{aligned} e_1 &= \mathrm{ReLU}(\boldsymbol{W}_1 e_{ui} + \boldsymbol{b}_1) \\ e_2 &= \mathrm{ReLU}(\boldsymbol{W}_2 e_1 + \boldsymbol{b}_2) \\ &\cdots \\ e_L &= \mathrm{ReLU}(\boldsymbol{W}_L e_{L-1} + \boldsymbol{b}_L) \end{aligned} \tag{6}$$

第 1 层输入的是池化层的输出 $\boldsymbol{e}_{ui}$, 每一层的输出通过 ReLU 函数变为下一层的输入。同时对每一层的输入赋予不同的权值 $\boldsymbol{W}_L$ 和偏置值 $\boldsymbol{b}_L$。

最后,深度交互层的输出 e_L 中富含丰富的高阶信息,使用一个线性回归的模型将 e_L 映射成最后的输出的预测得分:

$$y_{ui} = \boldsymbol{Z}^{\mathrm{T}} e_L + \boldsymbol{b}_u + \boldsymbol{b}_i \tag{7}$$

其中,$\boldsymbol{Z}$ 是权重向量,$\boldsymbol{b}_u$ 是用户偏置,$\boldsymbol{b}_i$ 是物品的偏置。然后通过一个点对损失函数对模型进行优化[25],从而最大程度地减少目标函数损失:

$$\begin{aligned} L(D \mid \boldsymbol{\Theta}) = \frac{-1}{\mid R^+ \mid + \mid R^- \mid} [& \sum_{(u,i) \in R^+} \log \delta(y_{ui}) \\ & + \sum_{(u,j) \in R^-} \log(1 - \delta(y_{uj}))] + \lambda_1 \| \boldsymbol{\Theta} \|^2 \end{aligned} \tag{8}$$

其中, D 是训练集,$\boldsymbol{\Theta}$ 是模型参数, R^+ 是正样本的集合, R^- 是负样本的集合;对于每一个正样本对,采取 4 个负样本对来进行匹配; $\delta(\cdot)$ 是一个激活函数,将输入的值映射到(0, 1)之间; $\| \boldsymbol{\Theta} \|^2$ 是 $L2$ 正则化参数,用来防止模型出现过拟合的情况,而 λ_1 则是用来控制正则化的学习率。

2.2 ARACF 模型

当人为地对模型输入添加一些微小扰动时,此时模型会将原始的输入样本识别成另一个样本,例如输入一个训练样本对 (u, i, j), 当添加扰动后,模型会将其识别成 (u', i, j), 从而导致模型性能下降。

为了增强 RACF 的抗干扰能力,提升模型的鲁棒性,使用对抗性学习的方法,对 RACF 进行对抗性训练。人为对模型输入添加扰动,通过 FGSM 算法截断地生成正态分布随机数以及梯度求导来构建对抗扰动。添加对抗扰动后的样本称为对抗样本,再将对抗样本和初始样本一起输入模型进行训练,使得模型尽可能多地见到更多的训练样本,最终得到具有良好鲁棒性的ARACF模型。在模型训练阶段,使用随机梯度下降法来训练模型直至收敛。因此对于 ARACF 来讲,其目标一方面要最小化训练样本的损失,另一方面要最大化对于参数扰动之后的训练样本的损失。在模型输入上添加的对抗性样本的扰动为

$$\Delta^* = \arg \max_{\Delta, \| \Delta \| \leqslant \varepsilon} L_{\mathrm{adv}}(D \mid \hat{\boldsymbol{\Theta}} + \Delta) \tag{9}$$

其中 Δ 是模型输入添加的扰动, ε 控制扰动的量级, $\hat{\boldsymbol{\Theta}}$ 是当前模型的参数。再对损失函数进行不断地梯度求导,直到找出一个使目标函数最大化的值,此时式(9)可由下面公式得到:

$$\Delta_{\mathrm{adv}} = \varepsilon \frac{\Gamma}{\| \Gamma \|} \leftarrow \Gamma = \frac{\partial L_{\mathrm{adv}}(D \mid \hat{\boldsymbol{\Theta}} + \Delta)}{\partial \Delta} \tag{10}$$

其中, ε 的值越大,对于模型的扰动也就越剧烈; $\| \cdot \|$ 是 l_2 的范数。从整体来看,上述问题可以解释为一个极大极小博弈问题,即在最小化目标函数的同时需要兼顾增加对抗性扰动而使得目标函数尽可能地最大化,最终达到一个均衡点,从而使得模型对于正常扰动的样本以及对抗性样本的鲁棒性都很强,如式(11)所示。

$$\boldsymbol{\Theta}^*, \Delta^* = \arg \min_{\boldsymbol{\Theta}} \max_{\Delta, \| \Delta \| \leqslant \varepsilon} L(D \mid \boldsymbol{\Theta}) + \lambda L_{\mathrm{adv}}(D \mid \boldsymbol{\Theta} + \Delta) \tag{11}$$

最后学习模型参数时,只需要最小化式(12)即可。

$$\begin{aligned} L_{\mathrm{ARACF}}(D \mid \boldsymbol{\Theta}) &= L(D \mid \boldsymbol{\Theta}) + \lambda L_{\mathrm{adv}}(D \mid \boldsymbol{\Theta} + \Delta_{\mathrm{adv}}) \\ &= \frac{-1}{\mid R^+ \mid + \mid R^- \mid} [\sum_{(u,i) \in R^+} \log \delta(y_{ui}) \\ &\quad + \sum_{(u,j) \in R^-} \log(1 - \delta(y_{uj})) \\ &\quad + \lambda (\sum_{(u,i) \in R^+} \log \delta(\hat{y}_{ui}) \\ &\quad + \sum_{(u,j) \in R^-} \log(1 - \delta(\hat{y}_{uj})))] \\ &\quad + \lambda_1 \| \boldsymbol{\Theta} \|^2 \end{aligned} \tag{12}$$

最终求得最优的模型参数以及扰动性较强的对抗样本。

2.3 算法流程

由于数据集中存在大量训练对,使用随机梯度下降(stochastic gradient descent, SGD)算法来加速模型的收敛,如算法 1 所示。

算法 1 ARACF-SGD 算法

输入：training data D, adversarial noise level ε, L2 regularize λ_1, learning rate η, Adversarial regularize strength λ,

输出：model parameter Θ

1. initialize Θ
2. while not converge do
3. get (u, i, j) from D
4. $\Delta_p \leftarrow \varepsilon \frac{\tau_p}{\| \tau_p \|}$ where $\tau_p = \frac{\partial L_{\text{adv}(D|\Delta_p)}}{\partial_{\Delta_p}}$
5. $\Delta_q \leftarrow \varepsilon \frac{\tau_q}{\| \tau_q \|}$ where $\tau_q = \frac{\partial L_{\text{adv}(D|\Delta_q)}}{\partial_{\Delta_q}}$
6. $p_i \leftarrow p_i - \eta \frac{\partial \text{loss}}{\partial \boldsymbol{p}_i}$
7. $q_j \leftarrow q_j - \eta \frac{\partial \text{loss}}{\partial \boldsymbol{q}_j}$
8. $\Theta = \Theta - \eta \frac{\partial L_{\text{ADICF}}(D \mid \Theta)}{\partial \Theta}$
9. iter + = 1
10. end
11. return Θ

3 实验与分析

3.1 数据集介绍

在 MovieLens-1M 和 Pinterest 这 2 个数据集上评估所提出的模型性能。Movielens-1M 是一个电影评价的数据集，被广泛用于协同过滤算法上，本文使用包括 100 万个评分、每个用户至少有 20 个评价的版本。Pinterest 是一个图片数据集，原始的 Pinterest 数据集非常稀疏，选择使用一个类似于 MovieLens-1M 的版本，即每个用户至少有 20 次交互。数据集信息如表 1 所示。

表 1 数据集信息

数据集	交互数	用户数	项目数	稀疏度
MovieLens-1M	100 209	3706	6040	4.47%
Pinterest	1 500 809	9916	55 187	0.27%

在实验中，从数据集随机选取 80% 作为训练集，用来训练模型；10% 用作验证集，用于调整模型参数；剩余 10% 作为测试集，用来评估模型性能。

3.2 评价指标

本文使用推荐系统常用的 2 个评价指标 *HR* 和 *NDCG* 来评价算法的性能。*HR* 的计算公式如下：

$$HR@K = \frac{numbersofHits@K}{|U|} \tag{13}$$

其中，$numbersofHits@K$ 是每个用户生成的 top-K 推荐列表中属于测试集的物品总和，$|U|$ 是测试集的物品总数。*NDCG* 的计算如下所示：

$$NDCG@K = \frac{\sum_{u \in U} NDCG_u@K}{|U|} \tag{14}$$

$NDCG_u@K$ 代表测试集中每个用户的 NDCG，可由式(15)得到。

$$NDCG_u@K = \frac{DCG_u@K}{IDCG} \tag{15}$$

其中，*IDCG* 是指理想排序下相关性高的物品排在前面的 *DCG*，$DCG_u@K$ 的计算公式如式(16)所示。

$$DCG_u@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{16}$$

其中，rel_i 表示物品在生成的推荐列表中处于 i 的相关性。

为了评估模型的有效性，将 ARACF 与以下基线模型进行实验对比。

Pop 是一种非个性化排名方法，先根据物品的交互次数来衡量其受欢迎程度，再根据受欢迎程度对物品进行排名。

K 近邻算法(k-nearest neighbor，KNN)是一种标准的基于物品协同过滤算法，通过选取 K 个最近邻来预测得分。

贝叶斯个性化排序-矩阵分解(Bayesian personalized ranking-matrix decomposition，BPR-MF)利用贝叶斯个性化排名损失来优化 MF 模型，广泛用于推荐算法中，通过内积运算来预测得分。

FISM 是一种基于物品的深度协同过滤推荐算法，学习低维的隐向量空间来捕获物品之间的关联相似度。

DICF 为基于物品的深度协同过滤算法，利用多层神经网络来对物品间的高阶关系进行建模。

参数设置：为了防止过拟合，将 L2 正则化因子的值设置为 $1e^{-6}$；实验中嵌入层大小设为 16；对于 ARACF 模型，先用 FISM 模型对其进行预训练从而达到更好的性能和较快的收敛速度；并使用均值为 0、标准差为 0.01 的高斯随机分布函数来初始化其

他的模型参数;学习率设为 0.01,超参数 a 的范围是[0, 0.1, …, 0.9, 1];至于多层感知网络,对其嵌入 3 层神经网络来探索物品间的高阶关系。

实验环境如下:Intel(R) Core(TM) i7-9700 CPU, NVIDIA GeForce RTX 2080Ti GPU, 16 GB 内存, Tensorflow-GPU 1.13.1,Python 3.6。

3.3 实验结果与分析

本节讨论了以下 3 个方面对模型性能的影响:(1)对比其他基线模型,比较它们的性能优劣;(2)添加了扰动之后的模型与原来模型的鲁棒性对比;(3)控制扰动的超参数。

实验 1 不同模型性能对比

本文比较了各个模型 Top-10 推荐的准确性,各模型的结果比较如表 2 所示。

表 2 各模型性能对比

数据集	Pinterest		MovieLens-1M	
算法	HR@10	NDCG@10	HR@10	NDCG@10
Pop	0.2652	0.1395	0.4518	0.2524
KNN	0.7553	0.5103	0.6242	0.3488
MF	0.8577	0.5385	0.6605	0.3872
FISM	0.8662	0.5523	0.6643	0.3949
DICF	0.8744	0.5612	0.6857	0.4115
RACF	0.8821	0.5638	0.6973	0.4246
ARACF	0.8903	0.5691	0.7086	0.4313

观察表 2 的数据可以得出以下结论。

(1)在 2 个数据集上,RACF 的表现都要优于 DICF,反映了不同物品的权重对推荐预测的影响,突出了探索用户对历史物品偏好的必要性,而注意力网络可以有效解决这一问题。

(2)加入了对抗性训练的 ARACF 在所有模型中表现最好,说明对抗性训练不仅可以提高模型鲁棒性,同时一定程度上也能够提升推荐模型的准确度。

(3)个性化推荐排名模型性能要远优于非个性化推荐,例如 Pop 和 BPR-MF 的得分对比,这也充分说明了个性化推荐的重要性。

(4)在高度稀疏的数据集上,与基于用户的协同过滤模型相比,基于物品的协同过滤模型往往能够取得更好的性能。

实验 2 模型鲁棒性对比

为了验证对抗性训练对模型鲁棒性的提升,将原来模型与经过对抗性训练的模型进行对比,通过对 2 个模型施加相同的扰动,从而比较它们性能的优劣,结果如表 3 所示。

表 3 模型鲁棒性对比

	Pinterest		MovieLens-1M	
扰动	RACF	ARACF	RACF	ARACF
$\varepsilon=0.01$	−8.2%	−2.4%	−15.2%	−3.3%
$\varepsilon=0.10$	−18.5%	−2.9%	−28.5%	−4.7%
$\varepsilon=1.00$	−44.3%	−8.5%	−60.1%	−13.6%

表 3 记录了不同 ε 的大小对模型影响的幅度。可以直观看出:当施加不同的扰动时,ARACF 总是比 RACF 要稳定,对扰动变得不敏感,这也充分说明了经过对抗性训练的模型鲁棒性有很大的提升。

实验 3 不同超参数对模型的影响

在 2 个数据集上进行实验,分别记录了迭代次数和 2 个扰动超参数 ε、λ 对模型的影响,为了研究某一参数的影响,当改变其中一个参数时将其余的参数保持不变。

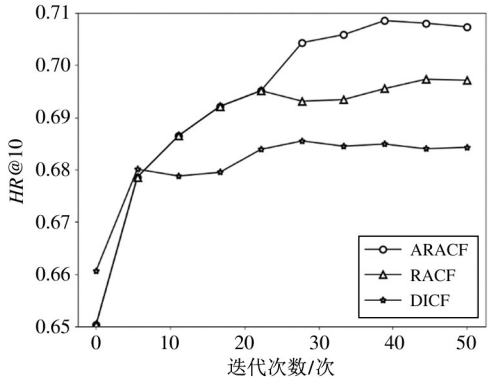
3.3.1 迭代次数的影响

在 MovieLens-1M 上的实验结果表明,迭代次数在 10 以内,DICF 的初始表现优于模型 RACF 和 ARACF。但随着迭代次数增加,DICF 过早地趋于拟合且性能较差,被本文提出的模型拉开较大差距。

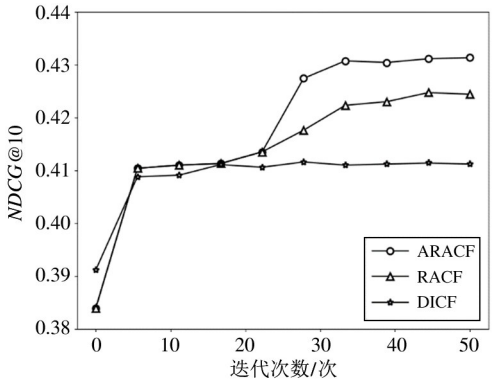
图 2 和图 3 表明,在 2 个数据集上,随着迭代次数 Epoch 增加,*HR*@10 和 *NDCG*@10 这 2 个指标整体上呈上升趋势,直至趋于稳定。可以看出,DICF 在训练过程中较早就达到了拟合状态且模型表现垫底。而加入了注意力机制的 RACF 模型性能明显要优于 DICF,这充分证明了注意力网络的有效作用。当迭代到 30 次后,继续对 RACF 进行训练,模型性能几乎没有任何提升,而 ARACF 却拥有较大的提升。这是在保持底层推荐模型保持不变,使用对抗性训练获得的结果。

3.3.2 超参数 ε 的影响

图4是ARACF中使用对抗性训练的超参数ε

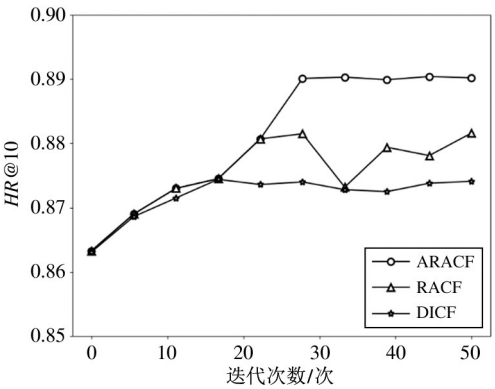


（a）HR 的训练曲线

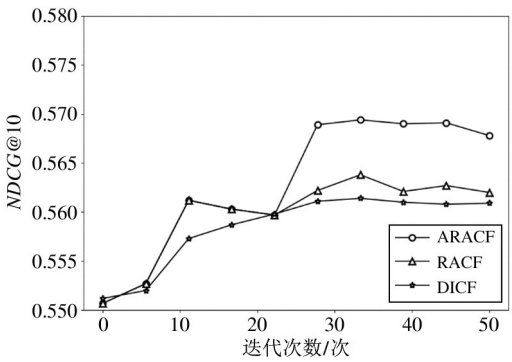


（b）NDCG 的训练曲线

图 2　在 MovieLens-1M 上的训练曲线对比

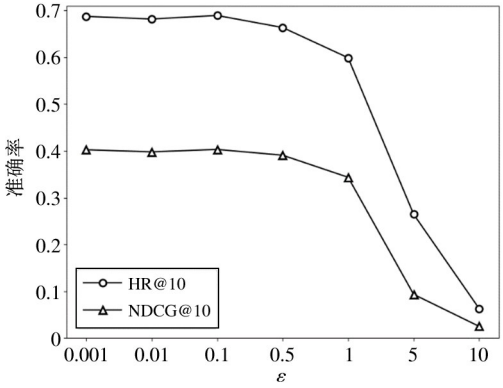


（a）HR 的训练曲线

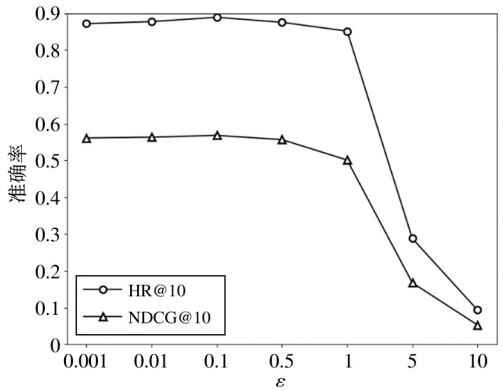


（b）NDCG 的训练曲线

图 3　在 Pinterest 上的训练曲线对比

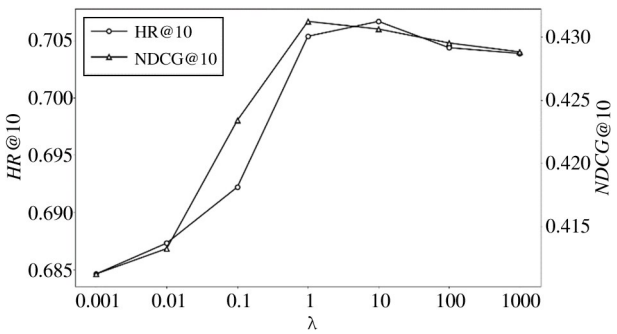


（a）在 MovieLens-1M 上的实验

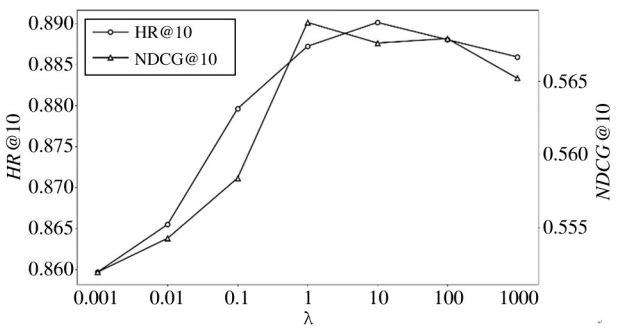


（b）在 Pinterest 上的实验

图 4　超参数 ε 的影响



（a）在 MovieLens-1M 上的实验



（b）在 Pinterest 上的实验

图 5　超参数 λ 的影响

对模型的影响。ε 控制着模型中扰动的大小,本文中 ε 的大小设为0.5。随着 ε 的增加,模型的性能整体上呈下降趋势。可以看出,在 ε =0.5 之前时,对性能几乎没有任何影响;当 ε 属于 0.5 ~1 时,性能缓慢下降;在 ε >1 后,性能直线下降;而当 ε 超过 10 时,曲线逼近0,这也说明过大的扰动会严重破坏模型的学习过程。

3.3.3 超参数 λ 的影响

λ 控制着对扰动的优化,可以看出:当 λ 小于1时,增加 λ 可以逐步改善模型的表现;当 λ 等于1时,模型达到最优性能。此时进一步增大 λ 既不会有效提升模型性能,也不会较大地降低模型的性能。这也就意味着过大的 λ 会使得模型对扰动变得不敏感。因此,λ 的大小在本文中设为1。

4 结 论

本文提出一种结合注意力机制的协同过滤推荐算法(RACF),通过注意力网络解决了推荐系统中用户不同历史物品的贡献度问题。在此基础上,为了研究推荐系统鲁棒性问题,提出了基于注意力机制的对抗性协同过滤推荐算法(ARACF)以此增强模型的鲁棒性。在2个数据集上进行的实验结果表明,本文所提算法不仅提升了模型的性能,还有效提升了模型的鲁棒性。未来将尝试2个研究方向:(1)利用用户和物品的附带信息(如频谱特征)和知识图谱来探索基于特征的推荐方法;(2)在对抗训练过程中,使用更优良的扰动构建算法来生成对抗样本。

参考文献

[1] ZANG F Z, YUAN N J, LIAN D F, et al. Collaborative knowledge base embedding for recommender systems[C]//The 22th ACM SIGKDD International Conference. New York: Association for Computing Machinery, 2016:353-362.

[2] 李昆仑,万品哲,张德智. 基于改进用户相似性度量和评分预测的协同过滤推荐算法[J]. 小型微型计算机系统,2018, 39(3): 567-571.

[3] WANG Z H, JIANG Z H, ZHAO C R, et al. A path-constrained framework for discriminating substitutable and complementary products in e-commerce[C]//Proceedings of the 11th ACM International Conference on Web Search and Data Mining. New York: Association for Computing Machinery, 2018:619-627.

[4] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms [C]// Proceedings of the 10th International Conference on World Wide Web. New York: WWW, 2001:285-295.

[5] NING X, KARYPIS G. SLIM: sparse linear methods for top-n recommender systems[C]//In 11th IEEE International Conference on Data Mining. Piscataway: IEEE, 2011:497-506.

[6] KABBUR S, NING X, KARYPIS G. FISM: factored item similarity models for top-n recommender systems[C]// Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2013:659-667.

[7] FENG X, HE X N, WANG X, et al. Deep item-based collaborative filtering for Top-N recommendation [J]. ACM Transactions on Information Systems, 2019,37(3): 1-25.

[8] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feedforward networks are universal approximators [J]. Neural Networks, 1989,2(5):359-366.

[9] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. California: NIPS, 2017:6000-6010.

[10] MOOSAVI-DEZFOOLI S M, FAWZI A, FAWZI O, et al. Universal adversarial perturbations[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017:86-94.

[11] GOODFELLOW I J, POUGET-ABADIE J, MEHDI M, et al. Generative adversarial networks[J]. Advances in Neural Information Processing Systems, 2014,3:2672-2680.

[12] LI H, LI J. Recognition of robot based on attention mechanism and convolutional neural network[C]//2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Chengdu: IEEE, 2019:2578-2584.

[13] XIAO J, YE H, HE X, et al. Attentional factorization machines: learning the weight of feature interaction via attention networks[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. Melbourne: Association for Computing Machinery, 2017: 3119-3125.

[14] ZHOU G R, ZHU X Q, SONG C R, et al. Deep interest network for click-through rate prediction[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: Association for Computing Machinery, 2018:1059-1068.

[15] SEO S Y, HUANG J, YANG H, et al. Interpretable convolutional neural networks with dual local and global attention for review rating prediction[C]// Proceedings of the 11th ACM Conference on Recommender Systems. Como: Association for Computing Machinery, 2017:297-305.

[16] YI T, LUU A T, SIU C H. Multi-pointer co-attention networks for recommendation[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: Association for Computing Machinery, 2018:2309-2318.

[17] HE X N, HE Z K, SONG J K, et al. NAIS: neural attentive item similarity model for recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2018,30(12):2354-2366.

[18] CHEN J W, ZHUANG F Z, HONG X, et al. Attention-driven factor model for explainable personalized recommendation[C]// Proceedings of the 41th International ACM SIGIR Conferenceon Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2018:909-912.

[19] HE X N, ZHANG H W, KAN M Y, et al. Fast matrix factorization for online recommendation with implicit feedback[C]//Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2016:549-558.

[20] HE X, HE Z K, DU X, et al. Adversarial personalized ranking for recommendation[C]//The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2018:355-364.

[21] TANG J, DU X, HE X, et al. Adversarial training towards robust multimedia recommender system[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 32(5):855-867.

[22] PARK D H, CHANG Y. Adversarial sampling and training for semi-supervised information retrieval[C]// The World Wide Web Conference. New York: WWW, 2019: 1443-1453.

[23] YUAN F, YAO L, BENATALLAH B. Adversarial collaborative auto-encoder for top-n recommendation[C]// International Joint Conference on Neural Networks. Budapest: IJCNN, 2019:1-8.

[24] YUAN F, YAO L, BENATALLAH B. Adversarial collaborative neural network for robust recommendation[C]// Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2019:1065-1068.

[25] BAI T, WEN J R, ZHANG J, et al. A neural collaborative filtering model with interaction based neighborhood[C]// Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. New York: Association for Computing Machinery, 2017:1979-1982.

Adversarial collaborative filtering recommendation algorithm based on attention mechanism

WU Zhefu, CHENG Jiebin, FANG Luping

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

Abstract

Aiming at the different contribution of items interacted by users to their decision-making in collaborative filtering recommendation algorithm, a collaborative filtering recommendation algorithm based on relevant attention is proposed. The algorithm combines the attention mechanism in deep learning to assign different weights to different items to capture the items most relevant to the target item, explore the influence of the weight of different items on the model prediction, and thereby improve the accuracy of the recommendation; further, in order to solve the problem of low robustness of the proposed model, a recommendation algorithm for attention collaborative adversarial training is proposed. Through the adversarial learning method and using the fast gradient sign method (FGSM) algorithm to build an adversarial sample input model for adversarial training, the model is alleviated from the impact of disturbance and the robustness of the model is improved. Experimental results on the two datasets of Pinterest and MovieLens-1M show that the proposed algorithm not only effectively improves the accuracy of the recommendation algorithm, but also enhances the robustness of the recommendation system.

Key words: collaborative filtering, attention mechanism, adversarial learning, robustness